CMSC 203, Assignment 4
Fall 2015

Concepts tested by this program

Aggregation,
Passing to and returning objects from methods
ArrayList Structure
Copy Constructor
Create a GUI driver class
- Stage
- Pane
- Event programming
- Textfields
- Labels
- Buttons
- Layouts (BorderPane, VBox,HBox)
- JOptionPane. showMessageDialog
- Read from a file

Deliverables:

Java files (source code)
Word Document with screenshots of test cases
JavaDoc Files

Deliverable format: The above deliverables will be packaged as follows. Two compressed files in the following formats:

- LastNameFirstName_Assignment4_Complete.zip, a compressed file containing the following:
-

  Word docuemts
  doc [a directory]
  - file1.html (example)
  - file2.html (example)

  src [a directory]
  - File1.java (example)
  - File2.java (example)

- LastNameFirstName_Assignment4_Moss.zip, a compressed file containing only the following:
  File1.java (example)
  File2.java (example)

Overview

The *iPic* Theater shows most popular new movies. It keeps a database of movies that contains information about movies and their current week sale. *iPic* Theater also keeps a record of previous week's sale for the SAME movies in a different database to compare sales for two weeks.

Write a Data Element class named *Movie* that holds the data for a movie. There will be a movie *title, rate (PG, PG13, R), 10 viewers review* (numbered 1-5) and the number *of sold tickets* for the current week.

Write a Data Manager Class named *MovieTheater* that holds a list of movies in an ArrayList. It also holds the average of reviews for each movie in a separate ArrayList. **Note that average list is created at the same time when movies list is created**. This class will have methods to load movies information from a file and store it in an Arraylist as well as their average, find the list of movies with highest and lowest reviews and the total sale. It will also have a method to compare current sale for each movie with a previous sale and report the list of movies that their sale has been increased.

Write a GUI class that allows the user to load movies information from a file, show highest and lowest average review, compare sale and find total sale.

Operation

When application starts, user is presented with a GUI to show movies information:

User can select *Load Movie* and select the file containing movies information. The application then displays movies' name, rating, reviews, average reviews and #of sold tickets.

When user selects *High/Low Average*, the list of movies with the highest and lowest average review will be displayed.

When user selects *Compare* and chooses the file containing movies' previous week, the list of movies that their sales has been increase will be displayed.

Total Sale will calculate the total sale.

Exit will exit the application.

Specifications

Data Element -*Movie*
The class *Movie* will contain:

1. Instance variables for movie title, rating, **an array** of int of 10 viewer's reviews (numbered 1-5) and number of sold tickets.
2. Method **calAvgReview** to calculate and return average of reviews.
3. Constructors (a copy constructor and parameterized constructor) and getter and setter methods.

Data Structure – An ArrayList of *Movie* to hold movie information and an array list of *Double* holding the average reviews for each Movie

Data Manager – *MovieTheater*, This class should not have any output functionality (e.g., no GUI-related or printing related functionality), but should take input, operate on the data structure, and return values or set variables that may be accessed with getters. **An instance of this class should be created in the GUI and applicable methods called to execute the program.**

The class *MovieTheater* will contain:

1. Instance variables of **an ArrrayList** of Movie and an **ArrayList** of Double.
2. Method **addMovies** – Pass in a parameter of type *Movie*. It will add the movie it to the list of movies. It also saves the average review for each movie in the ArrayList of Double.
3. Method **highestAndLowestReview**– Returns a string showing the list of movies with the highest and lowest average review.
4. Method **compareSales**- Pass in an ArrayList of Double, returns a String showing the list of movies which their sale have increased by comparing each movie's sale with the corresponding number in the array list passed in.
5. Method **getTotalSale**() – returns total number of sale.

You may need additional methods to include in this class.

User Interface – Contains instance variables for **labels and textboxes, buttons**… and an instance variable of *MovieTheater* :

1. Implements the EventHandler to listen to the GUI buttons.
2. When the *Load Movies* button is clicked the movies information is read from a file and displayed in the appropriate text boxes .Use a FileChooser to allow the user to select the file to read from. The file will contain the following information for each movie on a separate line:

   Title
   Rating
   10 user reviews (separated by space)
   #of sold tickets
3. When the *High/Low Reviews* button is clicked, a dialog box will show the list of the movies with highest and lowest reviews by calling *highestAndLowestReview***()** method from the Data Manager class.

4. When the *Compare* button is clicked, a FileChooser will allow the user to select the file that contains # of sale for the previous week for each movie. This file will have the following format:

   Movie Name

   #of sold tickets

   It will call *compareSales*() of the manager class and will display the list of the movies that have an increase in their sale amount.

5. *Total Sale* button, will set total of sales text box by calling *getTotalSale*() of the manager class.

- All the buttons except Load *Movies* and *exit* should be disabled at the beginning. Once information is loaded they will be enabled.

- All the text boxes are non-editable.

- Assume that the file format is correct.

File containing movie information:

currWeekSale.txt - Notepad

File Edit Format View Help

```
Terminator Genisys
PG13
5 5 5 4 4 3 5 2 1 1
5000
Ted2
R
5 4 4 4 4 5 5 5 5 3
15000
Minions
PG
5 4 4 4 4 5 5 5 5 3
60000
Jurassic World
PG13
5 4 5 4 4 3 5 2 1 1
10000
Inside Out
PG13
5 5 5 4 4 3 5 2 2 2
5000
MAx
PG
2 5 4 4 4 3 3 2 1 1
50230
SPY
R
5 3 5 4 4 3 5 2 1 1
3000
Me and Earl and the Dying Girl
PG13
5 2 5 4 1 3 5 2 1 1
1233
Mr. Holmes
PG
3 5 2 3 1 5 4 3 4 3
4500
Amy
R
3 5 2 3 1 5 5 5 4 3
20000
```

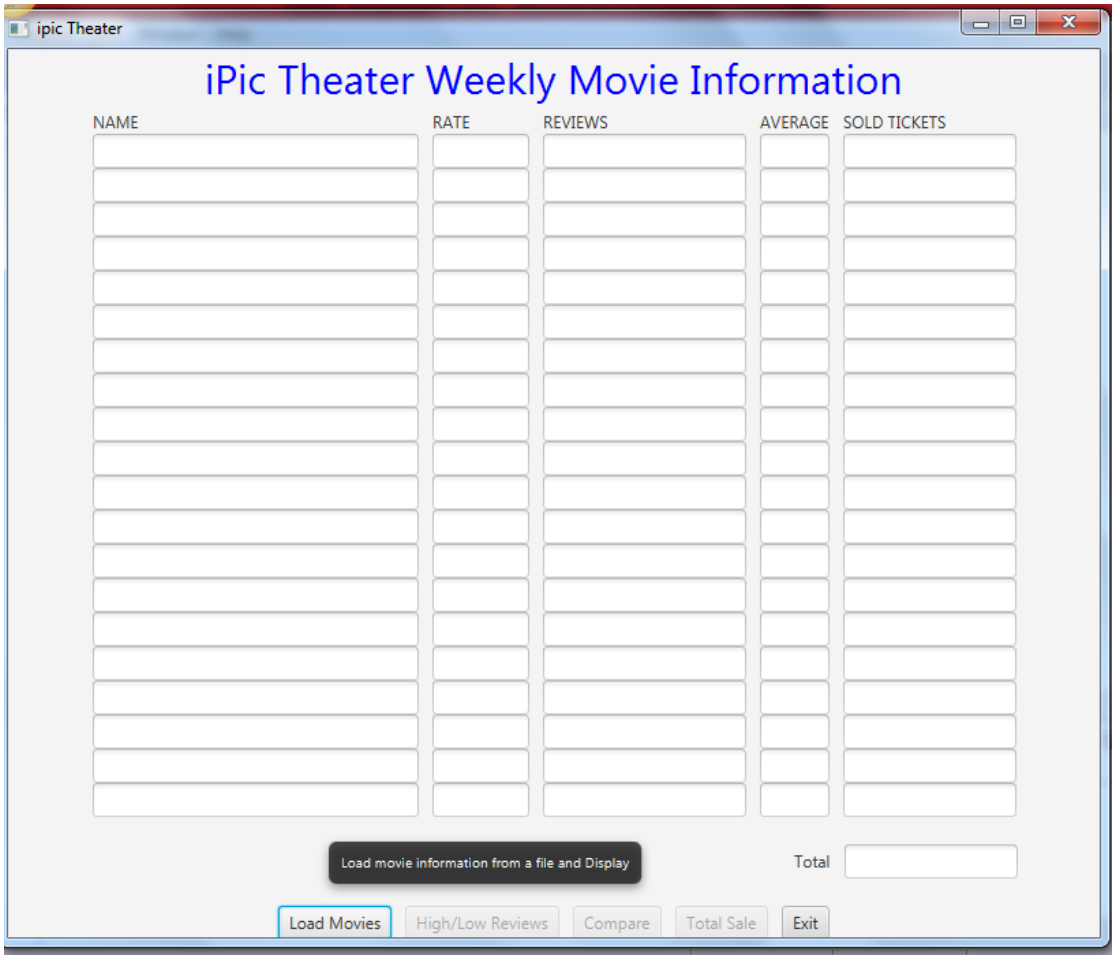
File containing previous week sale:

prevWeekSale.txt - Notepad

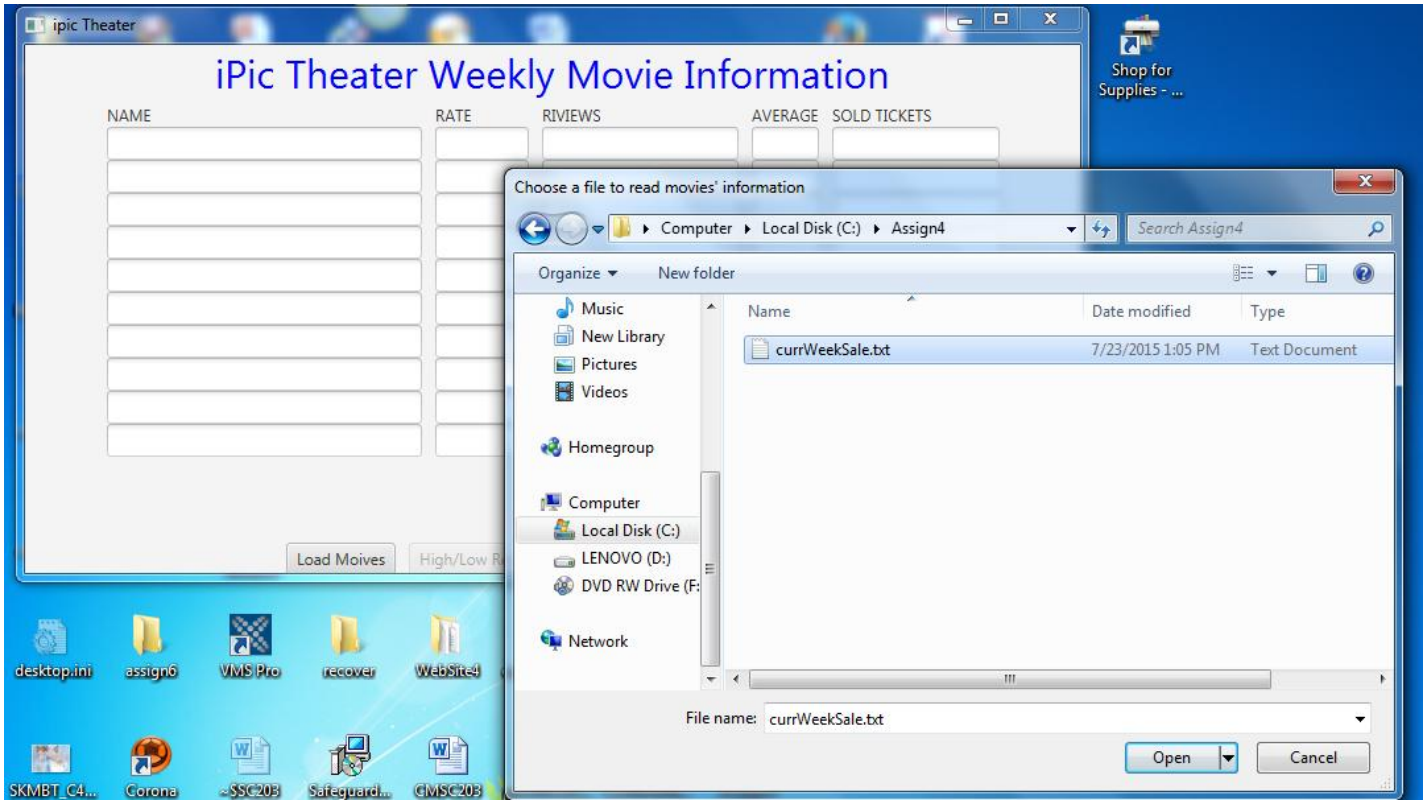
File Edit Format View Help

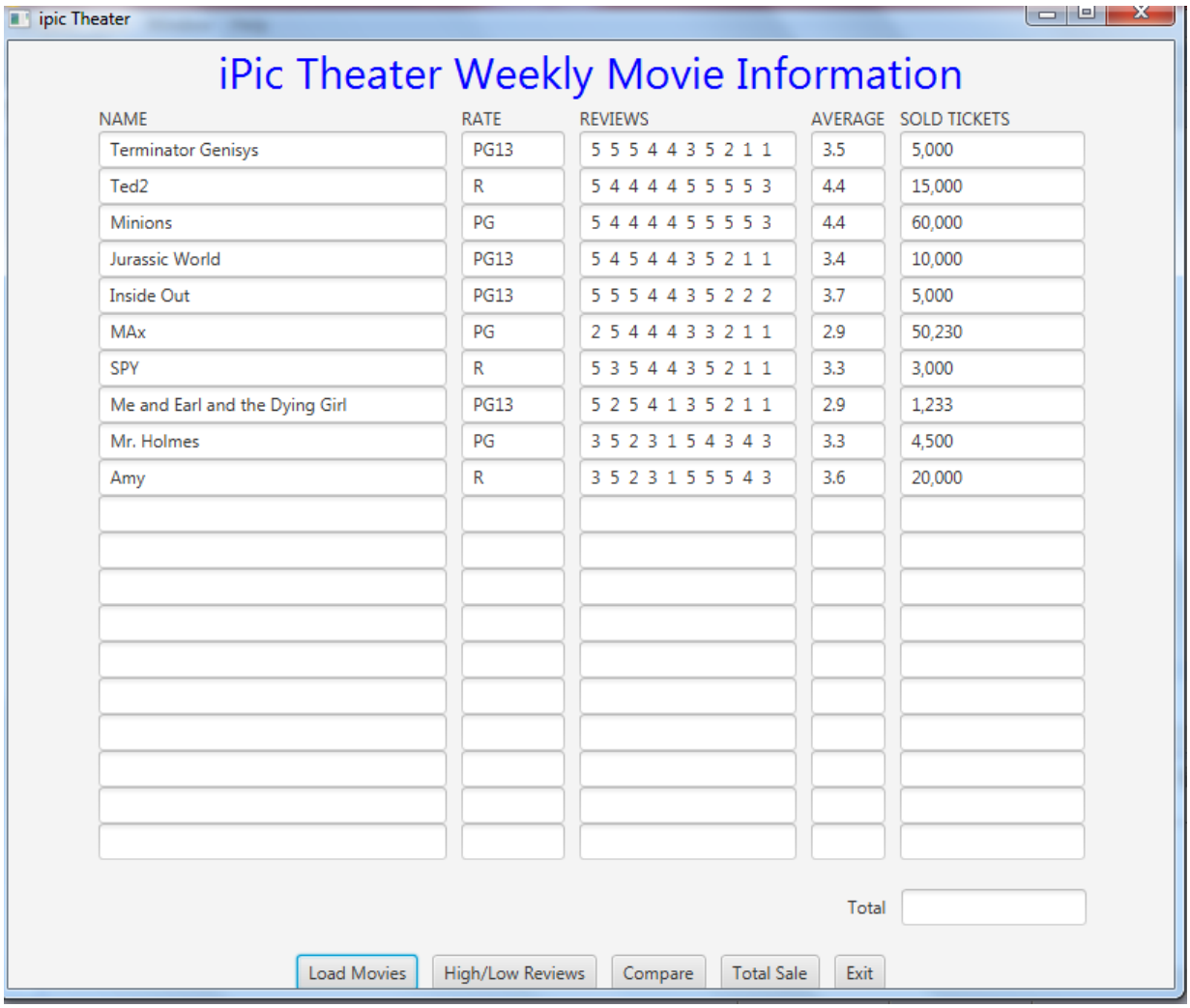
```
Terminator Genisys
3000
Ted2
20000
Minions
56000
Jurassic World
11000
Inside Out
5500
MAx
51000
SPY
2450
Me and Earl and the Dying Girl
1300
Mr. Holmes
4000
Amy
20000
```

When application starts:

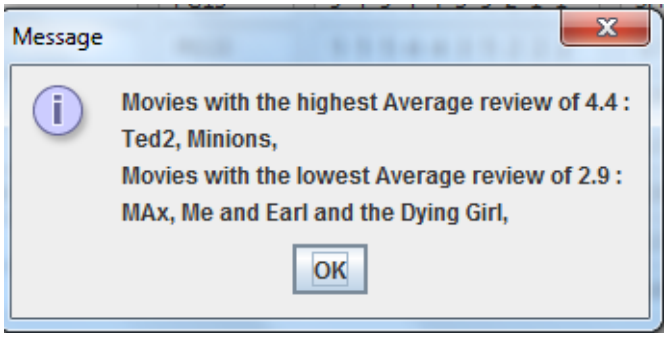


When Load Movies is selected:

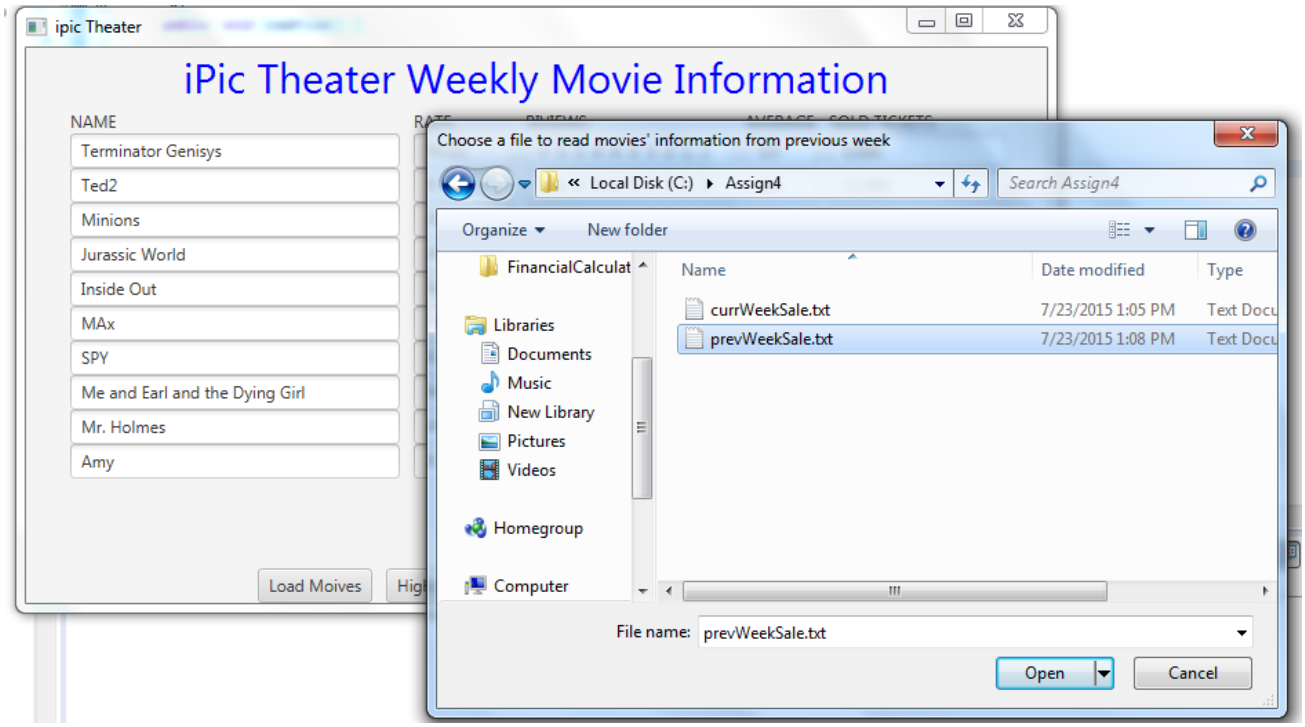


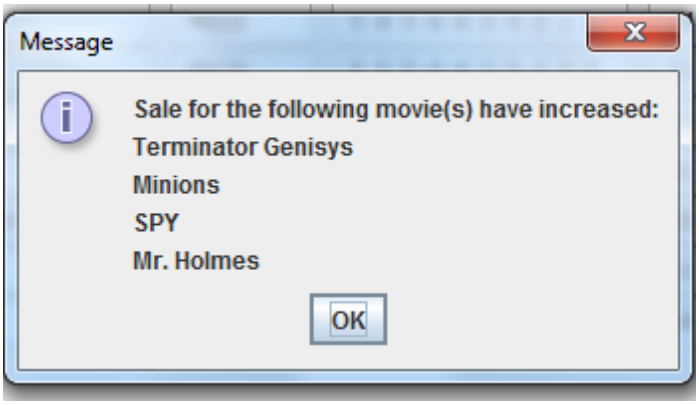


When High/Low review is selected:

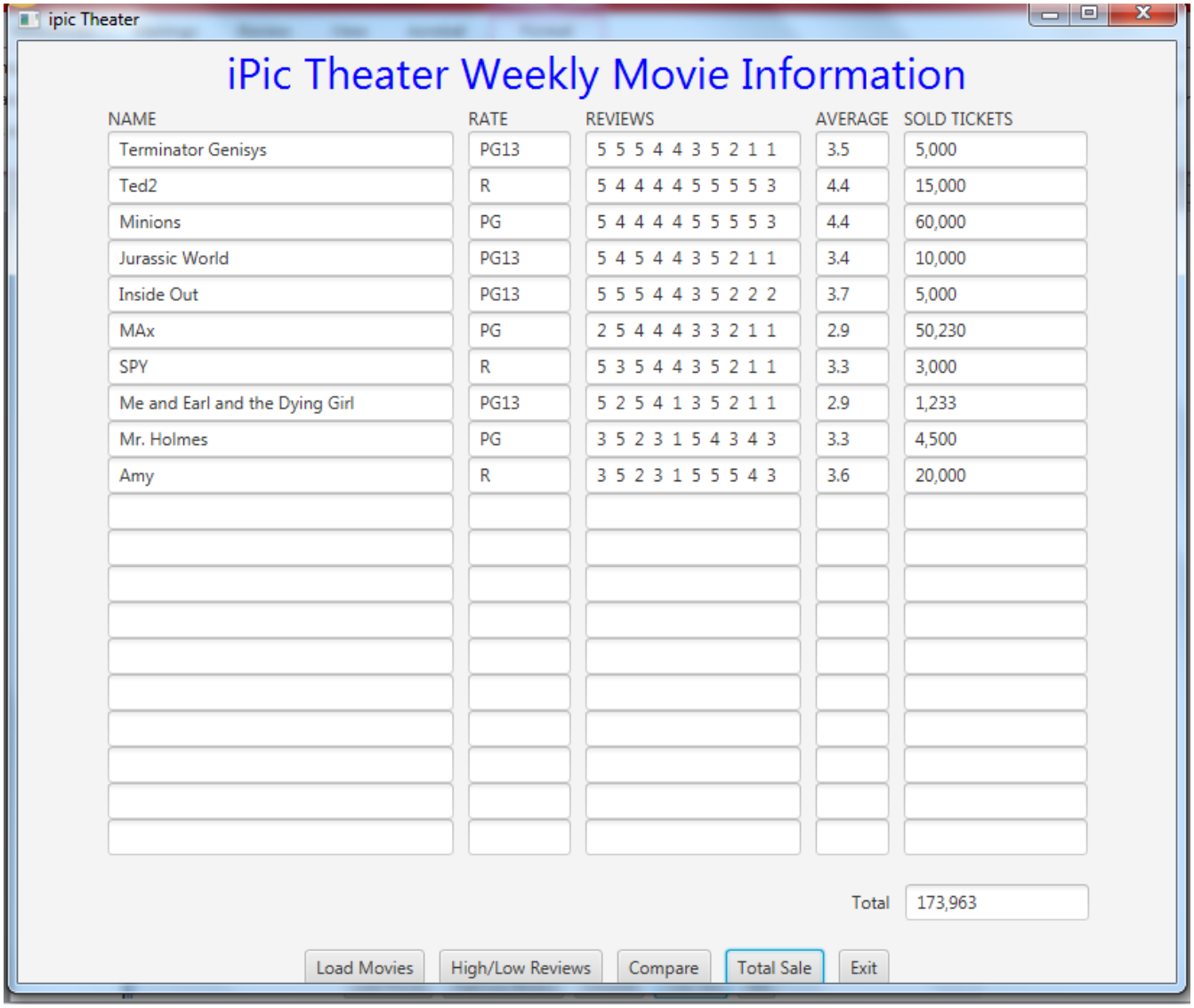


When Compare is selected:





When Total Sale is selected:



Program Grade Sheet Assignment #4

Name _____

DOCUMENTATION — Total 25 points
Javadoc for all user created classes — 7 pts _____

You have been asked to use an ArrayList for holding Movie information in DataManage(MovieTheater) class. Can you use a regular array instead? What are the Pros and Cons of using ArrayList over an Array? How would you change your interface to benefit from ArrayList structure? — 5 pts _____
(Include your answer in the word document containing your test cases' screen shots)

Test your application with 3 files having different number of movies. — 10 pts _____
Do not exceed 20 movies.

PROGRAMMING — Total 75 points

- Internal class Javadoc documentation (within source code) — 5 pts _____
  - Description of what each class does
  - Author's Name, @author
  - Methods commented properly using Javadoc documents
    - Description
    - @param, @return

- Compiles and Runs without runtime errors or warnings — 10 pts _____
- Program user interface
  - Clear to user how to use buttons — 3 pts _____
  - Output is easy to understand — 3 pts _____
- Accuracy
  - Public tests – files given to you to read data from — 10 pts _____
  - Private tests – other tests run by the instructor — 10 pts _____
- Program Details
  - Data Element - Movie — 12 pts _____
    1. Instance variables
       Title, rate, array of int for reviews, # of sold tickets
    2. Methods: constructors, getters and setters,
       method to calculate average of reviews
  - Data Manager – MovieTheater — 12 pts _____
    1. Instance variables
       ArrayList of Movies
       ArrayList of Double
    2. Methods addMovies, highestAndLowestReview, compareSales, getTotalSale
  - GUI – MvGui — 10 pts _____

1. Uses an object of the MovieTheater
2. Use of arrays to hold text fields
3. Load Movies, High/Low Review,Compare, Total Sale and Exit buttons handled by implementing EventHandler interface.
4. Uses the FileChooser to read from files
5. Uses DataManager class (MovieTheater) methods to send and receive data
6. At the start of the application only Load Movies and Exit buttons are enabled
7. Text fields are non-editable.

Total — 100 pts _____